

Lissage exact dans les arbres aléatoires triplets à sauts markoviens

WOJCIECH PIECZYNSKI¹, FABIEN SALZENSTEIN²¹ Laboratoire Communications, Images, et Traitement de l'Information
9, rue Charles Fourier, 91 000 Evry, France² Institut d'Électronique du Solide et des Systèmes
23, rue du Loess, 67037 Strasbourg Cedex 2, France¹Wojciech.Pieczynski@it-sudparis.eu²salzenst@iness.c-strasbourg.fr

Résumé – Dans le problème de lissage à saut considéré on considère trois processus X, R, Y, avec X, Y à valeurs continues et R à valeurs discrètes. Y modélise le signal observé, X le signal caché que l'on cherche à estimer, et R les « changements de régime », ou des « sauts » du système. Dans le cas des processus mono-dimensionnels, où les processus en présence sont des chaînes, les modèles classiques sont des systèmes linéaires à sauts markoviens : R est une chaîne de Markov et la loi du couple (X, Y) conditionnellement à R est celle d'un système linéaire Gaussien présentant des facilités de calcul. Cependant, lorsque R n'est pas connu, les calculs ne peuvent plus être faits avec une complexité raisonnable et on doit faire appel à des approximations. Des modèles différents ont été proposés récemment, autorisant le lissage avec une complexité linéaire en temps. L'objet du présent article est d'étendre ces modèles aux arbres de Markov triplets.

Abstract – The problem of smoothing in switching linear systems dealt with in the paper consists of considering three stochastic processes X, R, Y, with X, Y continuous and R finite. Y models the observed signal, X models the hidden searched signal, and R models the “changes of regime”, or « switches », of the system. In the mono-dimensional case, where the three processes are random chains, the classical models consists of taking a Markov distribution for R, and a linear Gaussian system distribution for the distribution of (X, Y) conditional on R. Then, when R is known, smoothing can be performed with a reasonable complexity. However, when R is not known, such models do not allow the smoothing with reasonable complexity. Different models allowing such computation have been proposed recently. The aim of this paper is to extend them to the triplet Markov trees.

1 Introduction

Considérons $X_1^N = (X_1, \dots, X_N)$, $R_1^N = (R_1, \dots, R_N)$, $Y_1^N = (Y_1, \dots, Y_N)$ trois processus stochastiques, avec X_1^N caché à valeurs dans R^q , R_1^N caché à valeurs dans un ensemble fini, et Y_1^N observé à valeurs dans R^m . Le processus R_1^N modélise les « sauts » de la distribution du couple (X_1^N, Y_1^N) . Le problème de lissage est celui du calcul, pour tout $n = 1, \dots, N$, de l'espérance conditionnelle $E[X_n | Y_1^N = y_1^N]$, qui sera notée $E[X_n | y_1^N]$. Le modèle probabiliste classique gaussien [3, 4, 10] consiste à poser :

$$R_1^N \text{ est une chaîne de Markov ;} \quad (1)$$

$$X_n = F_n(R_n)X_{n-1} + G_n(R_n)W_n ; \quad (2)$$

$$Y_n = H_n(R_n)X_n + J_n(R_n)Z_n, \quad (3)$$

avec (W_n) et (Z_n) des variables indépendantes gaussiennes et $F_n(R_n)$, $G_n(R_n)$, $H_n(R_n)$, $J_n(R_n)$ des matrices de tailles correspondantes et dépendantes de R_n . Conditionnellement à R_1^N , la loi du couple

(X_1^N, Y_1^N) est alors celle du modèle linéaire gaussien classique dans lequel le lissage est calculable [3, 10]. Il s'agit donc là d'une extension « naturelle » au modèle à sauts. Cependant, lorsque R_1^N dans le modèle (1)-(3) n'est pas observé, le calcul de $E[X_n | y_1^N]$ avec une complexité linéaire (ou même polynomiale) en N est impossible [2-4, 16]. Ce calcul devient possible dans les deux modèles récents suivants [1, 2, 14]. Dans le premier on garde (1), et (2)-(3) est remplacé par :

$$p(x_1^N | r_1^N, y_1^N) = \prod_{i=1}^N p(x_i | r_i) ; \quad (4)$$

$$p(y_1^N | r_1^N, x_1^N) = \prod_{i=1}^N p(y_i | r_i). \quad (5)$$

C'est un modèle très simple, aussi obtenu à partir de (1)-(3) en prenant $F_n(R_n) = H_n(R_n) = 0$. Dans le deuxième on pose :

$$(R_1^N, Y_1^N) \text{ est de Markov ;} \quad (6)$$

$$X_{n+1} = F_{n+1}(R_{n+1}, Y_{n+1})X_n + G_{n+1}(R_{n+1}, Y_{n+1})W_{n+1}, \quad (7)$$

avec (W_n) des variables indépendantes centrées (non nécessairement gaussiennes) et $F_{n+1}(R_{n+1}, Y_{n+1})$, $G_{n+1}(R_{n+1}, Y_{n+1})$ des matrices dépendantes de (R_{n+1}, Y_{n+1}) . Notons que la markovianité de (R_1^N, Y_1^N) - un tel modèle est dit « chaîne de Markov couple » - n'implique pas la markovianité de R_1^N ; les chaînes de Markov couples sont ainsi plus générales que les chaînes de Markov cachées et cette plus grande généralité se traduit par une plus grande efficacité des traitements non supervisés [5]. Les graphes de dépendance orientés des trois modèles ainsi obtenus, notés respectivement (a), (b), et (c), sont présentés à la Figure 1. On montre que le filtrage et le lissage exacts sont calculables dans les modèles (b) et (c) avec la complexité linéaire en N [13, 14], alors que de tels calculs ne sont pas possibles dans le modèle classique (a) [3, 4]. La raison profonde de cette différence, importante au plan applicatif, est la suivante. Le triplet $T_1^N = (X_1^N, R_1^N, Y_1^N)$ est de Markov dans les deux cas; cependant, dans le modèle (a) (X_1^N, R_1^N) est de Markov et (R_1^N, Y_1^N) ne l'est pas, alors que dans le modèle (c) la situation est inverse: (R_1^N, Y_1^N) est de Markov et (X_1^N, R_1^N) ne l'est pas. La markovianité de (X_1^N, R_1^N) ne présente pas d'utilité évidente, alors que la markovianité de (R_1^N, Y_1^N) permet de calculer $E[X_{n+1}|r_{n+1}, y_1^N]$ et $p(r_{n+1}|y_1^N)$ à partir de $E[X_n|r_n, y_1^N]$ et $p(r_n|y_1^N)$, ce qui mène à la possibilité de calculer $E[X_n|y_1^N]$ avec une complexité linéaire en N . En effet, en prenant l'espérance de (7) conditionnelle à $(R_{n+1}, Y_{n+1}) = (r_{n+1}, y_{n+1})$ on a

$$\begin{aligned} E[X_{n+1}|r_{n+1}, y_1^N] &= F_{n+1}(r_{n+1}, y_{n+1})E[X_n|r_{n+1}, y_1^N] = \\ &= \sum_{r_n} F_{n+1}(r_{n+1}, y_{n+1})E[X_n, r_n|r_{n+1}, y_1^N] = \\ &= F_{n+1}(r_{n+1}, y_{n+1}) \sum_{r_n} E[X_n|r_n, r_{n+1}, y_1^N] p(r_n|r_{n+1}, y_1^N) = \\ &= F_{n+1}(r_{n+1}, y_{n+1}) \sum_{r_n} E[X_n|r_n, y_1^N] p(r_n|r_{n+1}, y_1^N) \end{aligned}$$

La dernière égalité étant due à l'égalité $E[X_n|r_n, r_{n+1}, y_1^N] = E[X_n|r_n, y_1^N]$.

Nous avons donc

$$\begin{aligned} E[X_{n+1}|r_{n+1}, y_1^N] &= \\ &= F_{n+1}(r_{n+1}, y_{n+1}) \sum_{r_n} E[X_n|r_n, y_1^N] p(r_n|r_{n+1}, y_1^N) \end{aligned} \quad (8)$$

Finalement, les marginales $p(r_{n+1}|y_1^N)$ et les transitions $p(r_{n+1}|r_n, y_1^N)$ a posteriori - qui donnent

$p(r_n|r_{n+1}, y_1^N)$ - sont calculées par les démarches classiques dans les chaînes de Markov couples [5], ensuite $E[X_{n+1}|r_{n+1}, y_1^N]$ est donnée à partir de $E[X_n|r_n, y_1^N]$ par (8), ce qui donne également $E[X_{n+1}|y_1^N]$ par

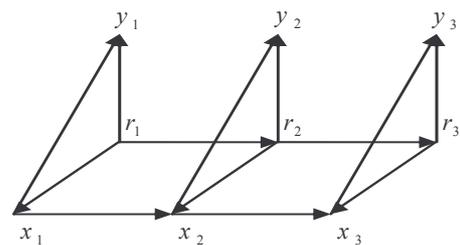
$$E[X_{n+1}|y_1^N] = \sum_{r_{n+1}} E[X_{n+1}|r_{n+1}, y_1^N] p(r_{n+1}|y_1^N) \quad (9)$$

Remarques

1. La différence entre les modèles (a) et (c) peut également être appréciée « localement », au niveau des transitions. Sachant que $T_1^N = (X_1^N, R_1^N, Y_1^N)$ est de Markov dans les deux cas, les différences entre les deux modèles doivent apparaître au niveau des transitions $p(t_{n+1}|t_n)$. Dans le cas classique (a) elles sont de la forme $p(t_{n+1}|t_n) = p(x_{n+1}, r_{n+1}|x_n, r_n) \times p(y_{n+1}|x_{n+1}, r_{n+1}, t_n)$, ce qui revient à considérer la simplification suivante: $p(x_{n+1}, r_{n+1}|x_n, r_n, y_n) = p(x_{n+1}, r_{n+1}|x_n, r_n)$. Dans le cas du modèle (c) on a $p(t_{n+1}|t_n) = p(r_{n+1}, y_{n+1}|r_n, y_n) p(x_{n+1}|r_{n+1}, y_{n+1}, t_n)$, ce qui revient à considérer la simplification suivante: $p(r_{n+1}, y_{n+1}|x_n, r_n, y_n) = p(r_{n+1}, y_{n+1}|r_n, y_n)$. Il ne nous semble pas, a priori, que l'une de ces simplification soit plus aisée à justifier que l'autre;

2. Le modèle (b) est simultanément un cas particulier du modèle (a) et du modèle (c). La deuxième propriété permet les calculs avec une complexité acceptable, et on peut alors considérer les résultats obtenus avec le modèle (b) comme une approximation des résultats qui seraient obtenus avec le modèle (a). Sachant qu'il existe d'autres méthodes d'approximation fondées sur le filtrage particulière, il est possible de les comparer avec celle utilisant le modèle (b);

3. Dans le modèle (c) le processus (R_1^N, Y_1^N) est de Markov, mais le processus des sauts R_1^N peut ne pas l'être. Par ailleurs, il est possible d'utiliser la démarche générale proposée dans [12] et utilisée dans le modèle simple (b) dans [1] pour rendre R_1^N semi-markoviens.



(a)

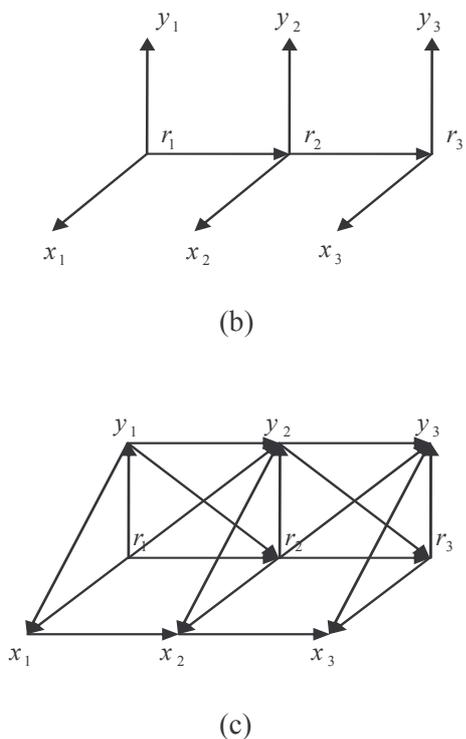


Figure 1 : graphes orientés de dépendance : modèle classique (a), modèle en peigne (b), modèle complet (c)

2 Arbres de Markov triplets

Les arbres de Markov cachés admettent de nombreuses applications, en particulier en traitement d'images et des signaux multi-résolution [6-9, 15, 17]. De manière générale les différentes techniques de calculs valables dans les chaînes de Markov cachées ou couples classiques se transposent relativement aisément aux arbres de Markov, et cela reste vrai lorsque l'on généralise le modèle (c) à une structure d'arbre.

Considérons un ensemble fini S muni d'une structure d'arbre orienté, et un processus triplet $T = (X, R, Y) = (X_s, R_s, Y_s)_{s \in S}$. Les X_s sont à valeurs dans R^q , les Y_s sont à valeurs dans R^m , et les R_s sont à valeurs dans un ensemble fini. Comme précédemment, (X, R) est caché, Y est observé, et le problème de lissage considéré est celui de calculer $E[X_s | y]$. On désignera le sommet par s_0 et pour tout $s \in S$ différent du sommet, on notera s^- le parent de s , que l'on supposera unique. Les variables indicées par les points ayant le parent commun sont supposées indépendantes conditionnellement à la variable indicée par ce parent. La loi d'un arbre de Markov triplet s'écrit alors

$$p(t) = p(t_{s_0}) \prod_{s \in S - \{s_0\}} p(t_s | t_{s^-}) \quad (10)$$

Ainsi la loi de $T = (X, R, Y)$ est donnée par $p(t_{s_0})$ les transitions $p(t_s | t_{s^-})$. En reprenant les trois types de transitions définissant les modèles (a), (b), et (c) de la section précédente nous obtenons trois arbres de Markov triplets.

Le premier modèle, noté (A), vérifie

$$R \text{ est un arbre de Markov ;} \quad (11)$$

$$X_s = F_s(R_s)X_{s^-} + G_s(R_s)W_s ; \quad (12)$$

$$Y_s = H_s(R_s)X_s + J_s(R_s)Z_s, \quad (13)$$

avec (W_s) et (Z_s) des variables indépendantes gaussiennes et $F_s(R_s)$, $G_s(R_s)$, $H_s(R_s)$, $J_s(R_s)$ des matrices dépendantes de R_s . Sachant que le modèle (A) est une extension du modèle (a), le calcul de $E[X_s | y]$ avec une complexité linéaire en $Card(S)$ n'est pas possible.

Le deuxième modèle, noté (B), vérifie

$$R \text{ est un arbre de Markov ;} \quad (14)$$

$$p(x, y | r) = p(r) \prod_{s \in S} p(x_s | r_s) p(y_s | r_s), \quad (15)$$

Ce modèle est simultanément une extension du modèle (b) et un cas particulier du modèle (C) défini ci-après.

Finalement, le modèle (C), qui étend le modèle (c) aux arbres, vérifie :

$$(R, Y) \text{ est un arbre de Markov ;} \quad (16)$$

$$X_s = F_s(R_s, Y_s)X_{s^-} + G_s(R_s, Y_s)W_s. \quad (17)$$

Nous allons montrer dans la section suivante qu'il est possible d'effectuer le calcul de $E[X_s | y]$ avec une complexité linéaire en $Card(S)$. Sachant que le modèle (B) est un cas particulier, ces mêmes calculs y seront alors applicables.

Un graphe de dépendance orienté d'un bi-arbre (chaque nœud admet deux enfants) est présenté à la Figure 2.

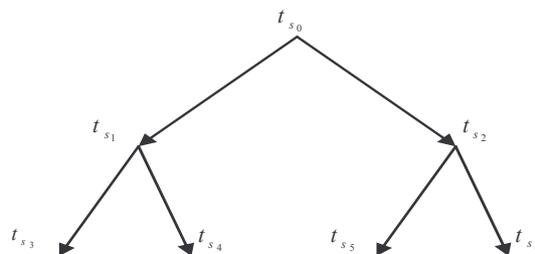


Figure 2: graphe orienté de dépendance d'un bi-arbre de Markov triplet

3 Lissage exact avec la complexité linéaire en Card(S)

Considérons un ensemble fini S muni d'une structure d'arbre orienté, et un processus triplet $T = (X, R, Y) = (X_s, R_s, Y_s)_{s \in S}$ dont la loi vérifie (16)-(17). De manière analogue aux calculs présentés dans la première section, on cherche à faire un lien entre $E[X_s | r_s, y]$ et $E[X_{s^-} | r_{s^-}, y]$. En prenant l'espérance de $X_s = F_s(R_s, Y_s)X_{s^-} + G_s(R_s, Y_s)W_s$ conditionnelle à $(R_{s^-}, Y) = (r_{s^-}, y)$ on arrive à

$$E[X_s | r_s, y] = F_s(r_s, y_s) \sum_{r_{s^-}} E[X_{s^-} | r_{s^-}, y] p(r_{s^-} | r_s, y), \quad (18)$$

où $p(r_{s^-} | r_s, y)$ sont données par les marginales $p(r_s | y)$ et les transitions $p(r_s | r_{s^-}, y)$, ces quantités étant calculées par les démarches classiques valables dans les arbres de Markov couples [11]. Ensuite $E[X_s | r_s, y]$ est donnée à partir de $E[X_{s^-} | r_{s^-}, y]$ par (9), ce qui s'écrit également

$$E[X_s | y_1^N] = \sum_{r_s} E[X_s | r_s, y_1^N] p(r_s | y_1^N) \quad (19)$$

4 Conclusion et perspectives

Nous avons étendu aux arbres de Markov triplets le récent modèle introduit dans le cadre des chaînes, autorisant le lissage exact avec complexité raisonnable dans des modèles à sauts. L'extension aux réseaux bayésiens plus généraux [10] fait partie des perspectives possibles pour les résultats présentés dans cet article.

Références

1. N. Abbassi and W. Pieczynski, Exact filtering in semi-Markov jumping system, Sixth International Conference of Computational Methods in Sciences and Engineering (ICCMSE 2008), September, 25-30, Hersonissos, Crete, Grece, 2008.
2. C. Andrieu, M. Davy, and A. Doucet, Efficient particle filtering for jump Markov systems. Application to time-varying autoregressions, *IEEE Trans. on Signal Processing*, Vol. 51, No. 7, pp. 1762-1770, 2003.
3. O. Cappé, E. Moulines, and T. Ryden, *Inference in hidden Markov models*, Springer, 2005.
4. O. L. V. Costa, M. D. Fragoso, and R. P. Marques, *Discrete time Markov jump linear systems*, New York, Springer-Verlag, 2005

5. S. Derrode and W. Pieczynski, Signal and image segmentation using pairwise Markov chains, *IEEE Trans. on Signal Processing*, Vol. 52, No. 9, pp. 2477-2489, 2004.
6. F. Desbouvries and J. Lecomte, Multiscale Bayesian restoration in pairwise Markov trees, *IEEE Trans. on Automatic Control*, Vol. 50, No. 8, pp. 1185-1190, 2005.
7. F. Desbouvries, J. Lecomte, and W. Pieczynski, Kalman filtering in pairwise Markov trees, *Signal Processing*, Vol. 86, No. 5, pp. 1049-1054, 2006.
8. J.-M. Laferté, P. Perez, and F. Heitz, Discrete Markov image modeling and inference on the quad-tree, *IEEE Trans. on Image Processing*, Vol. 9, No 3, 390-404, 2000.
9. E. Monfrini et W. Pieczynski, Estimation de mélanges généralisés dans les arbres de Markov cachés, application à la segmentation des images de cartons d'orgue de barbarie, *Traitement du Signal*, Vol. 22, No. 2, pp. 135-147, 2005.
10. K. Murphy, Dynamic Bayesian Networks, <http://www-clmc.usc.edu/publications/M/murphy-Chapter2002.pdf>, 2002.
11. W. Pieczynski, Arbres de Markov Couples, *Comptes Rendus de l'Académie des Sciences - Mathématique*, Série 1, Vol. 335, No. 1, pp. 79-82, 2002.
12. W. Pieczynski, Exact calculation of optimal filter in semi-Markov switching model, Fourth World Conference of the International Association for Statistical Computing (IASC 2008), December 5-8, Yokohama, Japan, 2008.
13. W. Pieczynski, Exact filtering and smoothing in short or long memory stochastic switching systems, 2009 IEEE International Workshop on Machine Learning for Signal Processing (MLSP 2009), September 2-4, Grenoble, France, 2009.
14. W. Pieczynski and F. Desbouvries, Exact Bayesian smoothing in triplet switching Markov chains, Complex data modeling and computationally intensive statistical methods for estimation and prediction (S. Co 2009), September 14-16, Milan, Italy, 2009.
15. J.-N. Provost, C. Collet, P. Rostaing, P. Pérez and P. Bouthemy, Hierarchical Markovian segmentation of multispectral images for the reconstruction of water depth maps, *Computer Vision and Image Understanding*, Vol. 93, No. 2, pp. 155-174, 2004.
16. J. K. Tugnait, Adaptive estimation and identification for discrete systems with Markov jump parameters, *IEEE Trans. on Automatic Control*, AC-25, pp. 1054-1065, 1982.
17. A. S. Willsky, Multiresolution Markov models for signal and image processing, *Proceedings of IEEE*, Vol. 90, No. 8, pp. 1396-1458, 2002.